The Value of Criticality: Gauging Issues in Supply Nets

Jochen K. SPEYERER
FORWIN (Bavarian Information Systems Research Network)
University of Erlangen-Nuremberg
Nuremberg, 90403, Germany

and

Andrew J. ZELLER
norisbank AG
Nuremberg, 90489, Germany

ABSTRACT

Modern day supply chains encompass both geographically disparate activities and planning processes for multiple companies or various interdependent time horizons. To be able to effectively manage these supply chains it is not only necessary to strategically plan the future of the underlying network of participating companies but also to schedule and monitor the ongoing production and logistics activities on a regular basis. Unfortunately, available information systems do not provide an adequate way to handle disruptions. If at all, they employ inter-organizational workflows to keep track of activities and notify a pre-set recipient in case something goes wrong. But in order to be able to focus their attention on urgent problems, managers need a means to gauge the criticality of a symptom. This paper tries to fill this gap by introducing a Value of Criticality (VoC) that indicates how serious the faced deviation really is.

Keywords: Disruption Management, Supply Chain Management, Flexible Integration, Supply Networks, Performance Measurement

1. INTRODUCTION

Background

The focus of modern production and supply processes has eventually shifted away from the single company to the competition of and within entire supply nets ([12]). To be able to effectively manage these supply chains it is not only necessary to strategically plan the future of the underlying network of participating companies but also to schedule and monitor the ongoing production and logistics activities on a regular basis using modern information technology to support a rigorous decision-making process.

To date, operational managers spend a substantial amount of their working time (up to 60 % [17]) on handling disruptions and unforeseen events in the network. In their study "The logistics footprint" well-known consulting firm Accenture characterized the ability to adequately measure the performance of a network and react to disruptions as a core capability [8]. Unfortunately, the problem has not been comprehensively addressed in literature. Moreover, managers nowadays face a steadily increasing amount of information, most of which is operational [23]. In addition to measuring performance and handling disruptions, each supply chain-oriented approach has to cope with network-specific problems: inherent instability due to reconfiguration, participation in more than one network at a time and the existence of small and medium sized enterprises (SME) that cannot afford expensive full-scale software solutions. Thus, no monolithic framework will satisfactorily fulfill these specific requirements. Instead, the approach will have to allow for companies to flexibly integrate into multiple networks.

Research Goal

The research conducted aims at finding a concept that allows networked organizations to quickly assess any operational disruption occurring within the entire net and assign a value of criticality (VoC) to it. This value is supposed to represent typical disruption-related criteria and must also be easily applicable in business practice.

Research Methodology

The research methodology applied is based on four pillars: literature review, expert interviews, verification with a practice partner and the development of a prototype (Table 1).

Methodology	Literature review	Discussions with industry experts	Verification with practice partner	Research prototype
Focus	State of the art analysis	Relevance of research, business requirements and additional insight into operations	Factors influencing reactions to disruptions, relevance of proposed concepts	Proof of concept

Table 1 Research methodology

Due to limited availability of research material on disruption handling in networks, the development of the presented framework has been both inductive and deductive. Thus, we decided to evaluate the capabilities of commercially obtainable supply chain event management (SCEM) products in addition to an in-depth literature review. Discovered gaps were discussed and verified in face-to-face meetings with a practice partner specialized in transportation management and the herein proposed concepts were derived. To demonstrate the principles work in practice, a research prototype has been built.

Previous Work on Supply Net Controlling and Disruption Management

Previous research on disruption handling, supply chain controlling and performance measurement has mainly adopted five distinct approaches:

- intra-company handling of disruptions
- decentralized management of supply nets and transports
- performance measures and controlling frameworks
- risk management
- supply chain event management

Despite the fact that supply chain managers spend a large amount of their working time on handling exceptions within the network this area has been almost neglected in academic research. Readily available literature on disruptions focuses on rescheduling job-shops and optimizing production activities within a single company [1]. Although detailed analysis and useful insight is provided, these approaches do not and, due to their very nature, cannot incorporate supply net-specific problems.

New concepts for decentralized management of networks usually employ agent technology to represent independent decisions of co-operating enterprises [5]. Although publications in this area reach from more generic methodologies to concrete applications of agents in marketplaces [30] contesting in optimization tournaments, the major shortcoming is either the lack of exception handling capabilities or the absence of an overall framework for disruption management.

Research on performance measurement and controlling focuses either on finding new taxonomies for grouping metrics that describe a supply chain into standardized categories [14]; [7] or applying the Balanced Scorecard to the cooperation [19]; [15]. However, these works merely look at one single node of a network and thus fail to describe “the big picture”. More advanced principles [21]; [9] build on the Supply-Chain Operations Reference Model (SCOR) that is on its way to a standard cross-industry process model. Although this may be seen as a first step into the right direction, it is not clear yet what indicators should be collected from each network participant, who is supposed to be in charge of measuring, where the original network strategy comes from and by what means it should be integrated into the economic targets of the chain.

Risk and uncertainty have been addressed by numerous authors. [31] points out that supply risk is the probability of an incident within the inbound flow of materials that leads to an inability to meet customer demand or even to a threatening of customer life and safety. [28] differentiate between the elements of loss, the significance of loss and the associated uncertainties. [6] combine categories of risk to a holistic assessment framework. In general, risk management approaches identify weak points in the network structure and try to compute a probability of failure. They do not assess the criticality of disruptions that already have happened.

Only recently have commercial software vendors like SAP®, Vigilance® and SeeCommerce® begun to conquer the market potential by offering so-called SCEM systems [11]. The revolutionary idea of these systems is to support the required transcorporate decision-making process in case of disruptions all the way from diagnostics to rigorous deployment of long-term therapeutic measures [13]. However, most available systems so far merely transfer the idea of workflow management to a network of companies. Each activity gets a status assigned that a central SCEM system monitors by comparing it against a network schedule. In case of any deviations an alert is issued and sent to a pre-set recipient [2]. Unfortunately, diagnostic and therapeutic modules still seem to be underdeveloped.

2. ELEMENTS OF THE CRITICALITY CONCEPT

In order to allocate scarce resources to those disruptions needing an immediate remedy it is necessary to set up a criticality concept. This notion of focusing on the most critical parts and processes of a network [4]; [10] is wide-spread in literature. Nonetheless, a clear concept when a problem gets serious is still missing.

However, before introducing this concept we will first describe the underlying high-level system architecture, the hierarchical process model that serves as a foundation and go on with explaining the dimensions of criticality.

High Level Systems Architecture

As we still consider the human decision maker to be at the forefront of managing companies or networks, respectively, an application is not supposed to be fully automated and deliver judgments and devise a therapy thereupon. Instead, we aim at providing the user with helpful information and make recommendations on what options he has [27].

However, it does not make a lot of sense to build a stand-alone solution that does not integrate with the already existing infrastructure of networked companies. A viable concept must hence easily connect to heterogeneous systems and be much more the missing link that adds value without adding too much overhead.

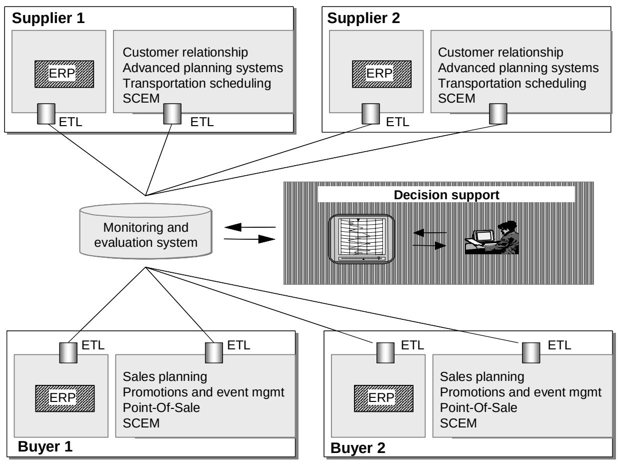

Figure 1 High-level architectural overview

Today, one may assume that most companies have installed enterprise resource planning (ERP) systems that manage and control their production lines and processes. Depending on the position in the supply chain, sophisticated software suites for demand planning, point-of-sale data retrieval, transportation planning, customer relationship as well as promotions and event management might be running.

All these systems and applications must interconnect to be able to leverage information sharing to the full extent (Figure 1). Thus, an extraction, transformation and loading (ETL) module needs to be set up at each site to transform streams of

transactional data into an agreed format. If all of the participants already have a working SCEM unit, we will only need to hook up to it and amalgamate the various messages into one central repository of the monitoring and evaluation system. For a more detailed explanation of the underlying data and component models the reader is referred to Speyerer and Zeller [24]; [25]; [26].
The next step consists of filtering the inbound disruption messages by their criticality. The following sections will present how this is accomplished.

Hierarchical Process Model
As already mentioned, a mandatory aspect in calculating the criticality value in case a disruption occurred is to have all necessary data about the process from the involved participants of the network on hand. Therefore, the monitoring system needs to model a process description out of the information provided by the ETL-module.

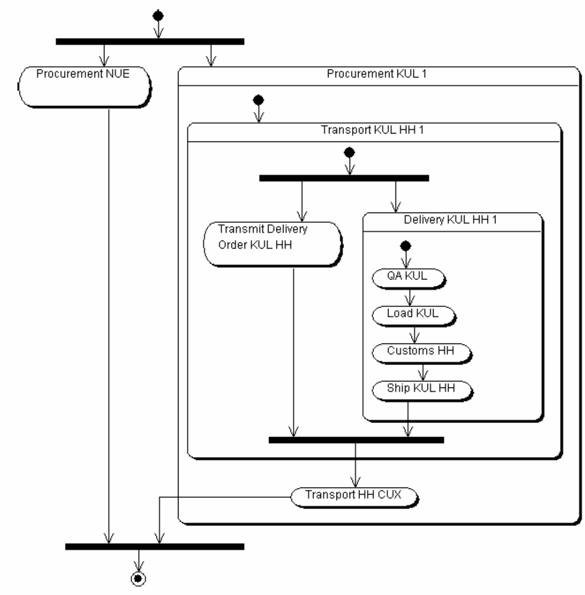

Figure 2 Example of a hierarchical process definition

In general, a process consists of several subprocesses which again can comprise multiple or single operations. The granularity of bundling activities to processes depends on the specific interest of the supply chain members and the system does not put any restrictions on that. Decomposing the ongoing production and logistics activities into small subprocesses has several advantages. Often accomplished operations, e. g. the transportation of goods to a warehouse and the following transshipment, as well as required document handling can be saved as subprocess and instantiated at any time without the need to remodel the definition the second time around.
Another benefit can be seen in the fact that the process definitions form a hierarchy. This allows for calculating time buffers between operations and to check whether these are sufficient to compensate for a disruption. Only in case that the duration of the delay exceeds this time span, the problem is escalated to the next layer in the hierarchy. Figure 2 illustrates an example of the just described hierarchical process model.

Dimensions of Criticality
Addressing the criticality problem requires a multi-dimensional approach. In other words, multiple factors need to be taken into account for a computation. The following list gives an overview:

1) Customer-oriented criticality: Using an ABC-evaluation the customer base is segmented into individual groups according to revenues generated. It is assumed that exceptions relating to A-customers are considered more serious.
2) Sector-oriented criticality: Supply networks may be split up into different sectors. The supply sector is comprised of all partners delivering raw materials. In this sector only little value-adding takes place. All manufacturing, producing and assembling is done in the production sector. The delivery processes of the final product to warehouses and the ultimate consumers are located in the distribution sector. Problems within the latter sector are supposedly more harmful as we are moving closer to the customer and thus have less time to steer clear to any disruptions.
3) Flow-oriented criticality: This dimension is based on graph theory [22]. Employing the already mentioned hierarchical processes model, an iteration algorithm checks on both available time and material buffers and determines if the impact of the disruption will reach through to the ultimate customer or can be compensated for by a node of the chain.
4) Situation-oriented criticality: Even problems that may seem harmless at first sight can escalate to critical disruptions in certain situations. A breakdown of a production machine should serve as an example. If one of several facilities breaks and the company faces average end customer demand, nothing serious is bound to happen as the producer switches to another machine or production line. Let us now consider an upcoming sales promotion with production running at full capacity. If the production machine breaks down this time and cannot be repaired for a number of days, the increased demand (stirred up by the promotional activities) cannot be satisfied.

3. PUTTING IT ALL TOGETHER: THE CONCEPT AT WORK

Now that the elements of the criticality concept have been introduced it is about time to join them in an integrative approach.

Required Data Input
To compute the criticality of a disruption we first need data concerning:

1) approximate number of units affected
2) estimated duration of disruption
3) layer of process model affected
4) process step where disruption occurred
5) type of product

These inputs are used as a reference point for finding out if time or material buffers along the supply chain can compensate for the missing or delayed units. A detailed explanation of the underlying architecture of the resulting prototype as well as other modules in the context of integrating the heterogeneous systems of networked organizations can be found in [24].

Adding a Scale to the Dimensions
Each of the previously mentioned dimensions must be assigned with an individual scale. This way, we can quickly measure problems and tell whether or not they will have a serious impact on the network and assign resources accordingly.

As the scale for Quality Function Deployment, introduced by Cohen [3] has been around for quite a while now, it makes sense to transfer it to our problem. A major advantage is its non-linearity which underscores the emphasis put on severe exceptions. Table 2 shows this concept for the customer dimension.

Scale value	Explanation	Application to dimension customer
1	Minor impact on network	C-customer affected
3	Moderate impact on network	B-customer affected
9	Serious impact on network	A-customer affected

Table 2 Scale for the customer dimension

The scale can be applied to the other dimensions (sector and situation) in a similar fashion.

Automated calculating of the flow-oriented criticality dimension

Before the system can set the value for flow-oriented criticality it will have to gauge the impact of the problem on the network. This can be accomplished by checking each process step for any available time or materials buffer. To do so, an algorithm iterates through the process model shown above. Figure 3, drawn using the Nassi-Shneiderman notation [18] illustrates the underlying logic. For the sake of clarity, only the main steps are shown in the diagram.

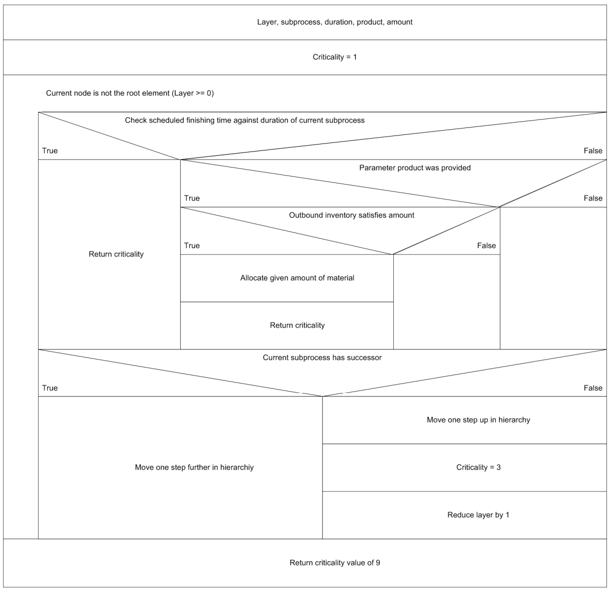

Figure 3 Simplified Nassi-Shneiderman diagram of the flow-oriented criticality algorithm

The first two parameters among the starting values stand for the current layer and the subprocess, where the disruption has been encountered. As long as the root element of the whole process has not been reached, the computation starts by calculating the estimated duration it takes for the current subprocess to finish. In case the buffer is not sufficient to compensate a delayed shipment the algorithm continues to check whether physical goods are concerned (missing freight documents would be an example of non-physical problems). In this instance, a product has been provided by the calling method and the function can continue testing if the outbound inventory located at the subsequent process step can satisfy the desired amount of the material affected. Should both aforementioned inquiries come up with a non satisfactory result, meaning that the element cannot remedy the problem by itself, we need to move one step further in the overall process model. Depending on whether the current subprocess has a task following or not, the computation starts all over, now with the successor or the parent of the already examined item. Once it gets necessary to ascend in the hierarchy of the process, the criticality is raised from 1 to 3. The algorithm stops at either a buffer big enough to compensate for the loss or the layer variable being smaller than zero. This means the problem cannot be solved before reaching the end-customer. The resulting value of the variable criticality is used to return the scale value for the flow-oriented dimension (Table 3).

Scale value	Position in hierarchy	Value of layer
1	Subprocess	Layer > 0
3	Different Layer	Layer ≥ 0
9	Root	Layer < 0

Table 3 Scale values for the flow-oriented criticality

We used C# as programming language and the Microsoft® .NET framework for developing the prototype. Figure 4 shows the implementation of the just described algorithm to calculate the criticality of a process. Some aspects, like error handling and necessary security checks, have been omitted.

```
private int calculateCriticality(int layer, Process subProcess,
                                 TimeSpan duration, Product product, int amount)
{
    int l = layer;
    int severeness = 1;
    Process currentProcess = subProcess;

    while(l >= 0)
    {
        // Check time buffer of current subprocess
        if(DateTime.UtcNow.Add(duration) <= currentProcess.getFinishingTime())
            return(severeness);
        // Check safety stock at destination if physical good is concerned
        else if(product != null)
            if(getFreeInventoryOutbound(currentProcess, product) >= amount)
            {
                allocateMaterial(currentProcess, product, amount);
                return(severeness);
            }

        // Move one step further in hierarchy or one step up
        if(currentProcess.hasSuccessor())
            currentProcess = currentProcess.getSuccessor();
        else
        {
            // Get parent process or null if root
            currentProcess = currentProcess.getParent();
            severeness = 3;
            l--;
        }
    }
    return(9);
}
```

Figure 4 Implementation of the criticality algorithm

Computing the VoC

Having chosen the scales, it is now possible to calculate the VoC for a disruption. The computation will be explained with

help of a fictitious example. Therefore, all the dimensions will be added to one scoring model.
In addition to the scale value of the individual dimensions, the network needs to achieve a consensus on the weights given to each of them. They should be selected according to their significance, which again is influenced by the overall market approach. In our example (Figure 5) the highest emphasis is placed on the customer dimension.

Dimension	Scale value	Weight	IM	TV	Status
Sector	3	30	90	>90	ok
Customer	9	40	360	>120	Yellow
Flow	1	20	20	>60	ok
Situation	1	10	10	>30	ok
Value of Criticality			**Σ 480**		

Figure 5 Exemplary computation of the VoC

Another decision requiring consent is devising the threshold value (TV) for the dimensions. The TV opens up the possibility to set up individual risk gates and distribute information not only based on the overall criticality measured by the VoC but also depending on the criticality level of a single dimension.
The scale value of each dimension is subsequently multiplied by its weight and results in the impact measure IM (this corresponds to the VoC per dimension). Both weights and TVs are exogenous to the model and must be determined by management.
Should the IM of a dimension be larger than the pre-set TV, the alert status will switch to yellow. Sticking to the customer criticality example, a key-account manager could be informed automatically in case an A-customer is affected.
By applying the just described concepts to a number of processes and activities of the network and taking the frequency into account, tasks which need a close supervision can be identified. Figure 6 shows the resulting portfolio. An in-depth discussion of this idea can be found in [29].

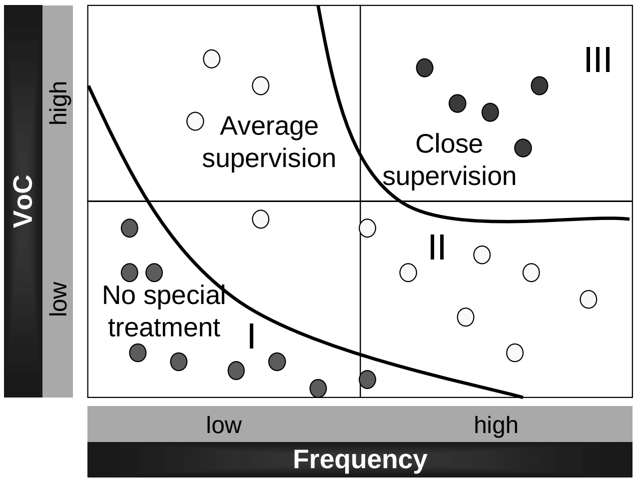

Figure 6 Classifying processes by VoC and frequency

4. CONCLUSION

The paper presents a multi-dimensional approach to measure the impact of operational disruptions in a networked organization. To give managers who spend a large part of their daily working time handling the effects of exceptions the possibility to make decisions based on a condensed measure instead of abundant unfiltered data, we developed an easy-to-use and easy-to-implement concept that calculates a value of criticality for every problem encountered.
As with any research there are many areas for improvement and future study, some of which include the automated presentation of the current criticality status of the overall network to the user and joining this operational analysis with a holistic medium and long-term controlling and measurement concept.
A futuristic, visionary goal for performance and disruption management could be an object-oriented supply network. Every item and product would know its destination and any handling restrictions. This way, e. g., a shipment of fresh, perishable fruits could contact the responsible retailer or logistics provider in case it senses it is getting too hot or even commence counteractions all on its own.

5. REFERENCES

[1] Belz, R. and Mertens, P., 1996. **Combining knowledge-based systems and simulation to solve rescheduling problems**, Decision Support Systems, 17 (2), 141-157.

[2] Bumstead, J., Ford, S. and Gillespie, R., 2003. **Integrated fulfilment**. In: Gattorna, John L. (Hrsg.): Gower handbook of supply chain management. 5th Ed. Gower. Burlington (USA).

[3] Cohen, L., 1995. **Quality Function Deployment: How to Make QFD Work for You.** Longman, Boston, MA.: Addison Wesley.

[4] Frederix, F., 2003. **Cooperation in Dynamic Networked Organizations.** In: Gasós, Jorge; Thoben, Klaus-Dieter (Eds.): E-Business Applications. Technologies for Tomorrow's solutions. Heidelberg: Springer.

[5] Garcia-Flores, R. and Wang, X. Z., 2002. **A multi-agent system for chemical supply chain simulation**

and management support, OR Spectrum 24 (3), 343-370.

[6] Harland, C., Brenchley, B. and Walker, H., 2003. **Risk in supply networks.** Journal of Purchasing and Supply Management, 9, 51-62.

[7] Hausman, W. H., 2003. **Supply Chain Performance Metrics.** http://www.stanford.edu/group/scforum/Welcome/Metrics121400R11.pdf. Download 2003-10-30.

[8] Hebert, L., 2001. **The Logistics Footprint: Creating a Road Map to Excellence.** In: Ascet 4 (2003) 1, 148 – 151.

[9] Hieber, R., 2002. **Supply Chain Management – A Collaborative Performance Measurement Approach.** VDF, Zurich /CH) 2002.

[10] Kanakamedala, K.; Ramsdell, G. and Srivatsan, V., 2003. **Getting supply chain software right.** In: The McKinsey Quarterly (2003) 1. Online Edition.

[11] Knickle, K., 2001. **Supply Chain Event Management – The Next Best Thing to Supply Chain Perfection**, AMR Research. AMR Research Outlook.

[12] Knolmeyer, G., Mertens, P. and Zeier, A., 2002. **Supply Chain Management Based on SAP Systems**, Order Management in Manufacturing Companies (SAP Excellence), Berlin Heidelberg: Springer.

[13] Lee, H. L. and Amaral, J., 2002. **Continuous and Sustainable Improvement Through Supply Chain Performance Measurement**, Stanford Global Supply Chain Forum. White Paper.

[14] Miller, T. ,2002. **Hierarchical Operations and Supply Chain Planning.** London: Springer.

[15] Morana, J. and Paché, G., 2003. **Building a Balanced Scorecard to Assess the Performance of Supply Chain Management – An Exploratory Study**, in: Pawar, Kulwant S.; Muffatto, Moreno (Eds.): Proceedings of the 8th International Symposium on Logistics. Nottingham (U.K): Centre for Concurrent Enterprise, 447-454.

[16] Morris, H. L., 2002. **Linking supply chain analytics to operations: the key to ROI.** IDC Viewpoint. Doc #VWP000086. February 2002.

[17] Mulani, N. and Lee, H. L., 2002. **New Business Models for Supply Chain Exellence.** In Ascet Volume 4, 14-18.

[18] Nassi, I. and Shneiderman, B., 1973. **Flowchart Techniques for Structured Programming.** ACM SIGPLAN Notices 8 (8), 12-26.

[19] Nuthall, L., 2003. **Supply chain performance measures and systems.** In: J.L. Gattorna, ed. Gower handbook of supply chain management. 5th ed. Burlington (USA): Gower, 248-266.

[20] Patig, S., 2001. **Using planning steps for production planning.** In: Proceedings of the International NAISO Congress on Information Science Innovations ISI, Dubai (U.A.E.).

[21] Pretko, P. 2001. **Supporting the Innovative SCOR Model with a Balanced Scorecard.** In: Ascet Vol.3 2001, 48-50.

[22] Sedgewick, R., 2001. **Algorithms in C++, Part 5: Graph Algorithms (3rd Edition)**, Boston: Addison Wesley Professional.

[23] Singh, S. K., Watson. H. and Watson, R., 2002. **EIS support for the strategic management process.** Decision Support Systems 33, 71-85.

[24] Speyerer, J. K., 2005. **Flexible Integration der Informationsverarbeitung im Supply Chain Management mit Web Services unter besonderer Berücksichtigung von Logistikdienstleistern.** Berlin: Dissertation.de.

[25] Speyerer, J. K. and Zeller, A. J., 2003a. **Integrating performance measurement with disruption handling for management support in supply nets.** In: Pawar, Kulwant S.; Muffatto, Moreno (Eds.): Proceedings of the 8th International Symposium on Logistics. Centre for Concurrent Enterprise. Nottingham (U.K.), 477-486.

[26] Speyerer, J. K. and Zeller, A. J., 2003b. **Disruption management in supply networks with web services - ad hoc information retrieval for fast impact analysis.** In: Proceedings of the 7th World Multiconference on Systemics, Cybernetics and Informatics. Volume I. International Institute of Informatics and Systemics. Orlando, FL (USA), 374-379.

[27] Wigand, R., Mertens, P., Bodendorf, F., Koenig, W., Picot, A. and Schumann, M., 2003. **Introduction to Business Information Systems.** Berlin, Heidelberg: Springer.

[28] Yates, J.F. and Stone E., 1992. **The risk construct.** In: J.F. Yates, ed. Risk Taking Behavior. New York: John Wiley & Sons, 1-25.

[29] Zeller, A. J., 2005. **Automatisierung von Diagnose und Therapie beim Controlling von Liefernetzen.** Aachen: Shaker.

[30] Zimmermann, R., Paschke, A., 2003. **PAMAS – an agent-based supply chain event management system.** In: Proceedings of AMCIS 2003, Mini-track on Intelligent Agents and Multi-Agent Systems, Tampa, 1892-1900.

[31] Zsisidin, G. A., 2003. A grounded definition of supply risk. Journal of Purchasing and Supply Management, 9,217-224.